M. Kedareswara Rao* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.5, Issue No.3, April – May 2017, 6018-6021.

Implementation of High-Speed Multiplier Filters Using a Modified Non Recursive Common Dada Multiplier

M. KEDARESWARARAO
Associate Professor, E.C.E Department
Avanthi Institute of Engineering and Technology,
Makavarapalem, Naraipatnam

PADATHALA KALPANA
Avanthi Institute of Engineering and Technology,
Makavarapalem, Naraipatnam

Abstract: **A multiplier is one of the key hardware blocks in most digital signal processing (DSP) systems. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis (Ayman.A et al (2001)). Many current DSP applications are targeted at portable, battery-operated systems, so that power dissipation becomes one of the primary design constraints. Since multipliers are rather complex circuits and must typically operate at a high system clock rate, reducing the delay of a multiplier is an essential part of satisfying the overall design.**

In this project two different multipliers are designed which are array multiplier and modified dada multiplier along with the combination of truncated multiplier. The comparison is carried out using the EDA tool XILINX ISE 12.3i by developing the RTL (Register Transfer Level) using the VERILOG HDL.

Keywords: **Truncated Multiplier; Array Multiplier; Modified Dada Multiplier; Multiplexer;**

I. INTRODUCTION

Multiplications are very expensive and slows the over all operation. The performance of many computational problems are often dominated by the speed at which a multiplication operation can be executed. Consider two unsigned binary numbers X and Y that are M and N bits wide, respectively. To introduce the multiplication operation, it is useful to express X and Y in the binary representation.

$$X = \Sigma Xi\ 2^i \quad i = 0 \text{ to } M$$
$$Y = \Sigma Yj\ 2^j \quad j = 0 \text{ to } N$$

$$Z = X \times Y = \Sigma Z_k 2^k \quad k = 0 \text{ to } M+N-1$$
$$= (\Sigma Xi\ 2^i \ i = 0 \text{ to } M)(\Sigma Yj\ 2^j \ j = 0 \text{ to } N)$$
$$= \Sigma(\Sigma X^i Y^j 2^{i+j}) \ i = 0 \text{ to } M\text{-}1, j = 0 \text{ to } N\text{-}1$$

The simplest way to perform a multiplication is to use a single two input adder. For inputs that are M and N bits wide, the multiplication tasks M cycles, using an N-bit adder. This shift –and-add algorithm for multiplication adds together M partial products. Each partial product is generated by multiplying the multiplicand with a bit of the multiplier – which, essentially, is an AND operation – and by shifting the result in the basis of the multiplier bit's position. Similar to the familiar long hand decimal multiplication, binary multiplication involves the addition of shifted versions of the multiplicand based on the value and position of each of the multiplier bits. As a matter of fact, it's much simpler to perform binary multiplication than decimal multiplication. The value of each digit of a binary number can only be 0 or 1, thus, depending on the value of the multiplier bit, the partial products can only be a copy of the multiplicand, or 0. In digital logic, this is simply an AND function.

A faster way to implement multiplication is to resort to an approach similar to manually computing a multiplication. The entire partial product are generated at the same time and organized in an array. A multi operand addition is applied to compute the final product. The approach is illustrated in the fig 1. This set of operation can be mapped directly into hardware. The resulting structure is called array multiplier.

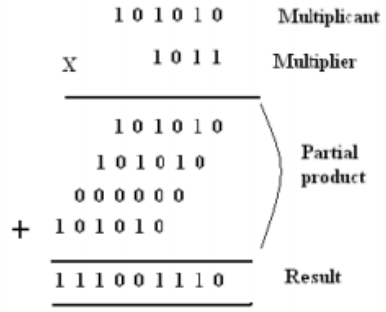


Fig:1 Example of manual multiplication

II. MULTIPLIER ARCHITECTURES

The composition of an array multiplier is shown in the Fig 2. There is a one to one topological correspondence between this hardware structure and the manual multiplication. The generation of n partial products requires N*M two bit AND gates. Most of the area of the multiplier is devoted to the adding of n partial products, which requires N-1,

M-bit adders. The shifting of the partial products for their proper alignment's performed by simple routing and does not require any logic. The overall structure can be easily be compacted into rectangle, resulting in very efficient layout.

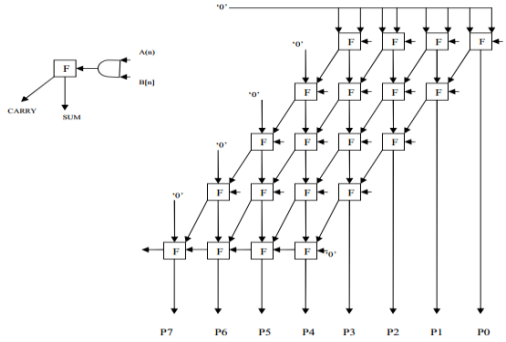


Fig: 2 Array Multiplier Architecture

Truncated multiplication is a technique which is shown in Fig 3 where only the most significant columns of the multiplication matrix are used and therefore area requirements can be reduced. In Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way, *T*, defines the degree of truncation and the *T* least significant bits of the product always result in *'0'*. The method of truncation will follows some steps in the process of multiplying of the partial product bits in the multiplier by the adders. The three steps involved in method are Deletion, Truncation, and Rounding.

In truncated multiplier we start the multiplication process with deletion only.

In the partial product bits we remove the more than half of the bits, and then remaining bits become the partial products in the process. This is the main criteria of deletion. Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way, T, defines the degree of truncation and the T Least Significant Bits (LSB) of the product always results in 0. The algorithm behind fixed width multiplication is the same as when dealing with non fixed width multiplication regardless of the truncation degree.

Conventionally an n-bit multiplicand and an n-bit multiplier would render a 2n-bit product. Sometimes an n-bit output is desired to reduce the number of stored bits. By the rounding process helps in the obtain of the faithfully rounded value. By these steps the truncated multiplier will gives the faithfully rounded values after truncate of the least significant part in the result. Truncated multiplication provides an efficient method for reducing the power dissipation and area of rounded parallel multiplier. The truncated multiplier is preferable as per the power related parameters, delay and area also it gives nominal results compare with the other multipliers. Truncated multiplier technique is an area reduced technique and it also it gives the low power values than the other one. With the truncated multipliers only the cost factor will be reduced in FIR filters.

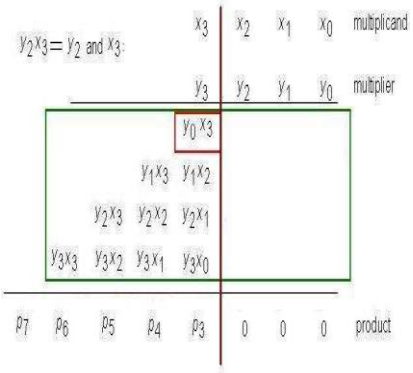


Fig:3 4x4 bit Binary Multiplication with truncation

A modified Wallace multiplier is an efficient hardware implementation of digital circuit which multiplies two integers whose flow chart is shown in Fig 4. Generally in the reduction phase of conventional Wallace multipliers, many full adders and half adders are used when compared to modified Wallace multipliers. As we know that half adders do not reduce the number of partial product bits. Therefore, it is necessary to minimize the number of half adders used in a multiplier which reduces the hardware complexity. Hence, a modification to the Wallace reduction is done in which the delay is the same as for the conventional Wallace reduction. The modified reduction method greatly reduces the number of half adders with a very slight increase in the number of full adders. Reduced complexity Wallace multiplier reduction consists of three stages. First stage the N x N product matrix is formed and before passing on to the second phase the product matrix is rearranged to take the shape of inverted pyramid. During the second phase the rearranged product matrix is grouped into non-overlapping group of three as shown below, single bit and two bits in the group will be passed on to the next stage and three bits are given to a full adder. The number of rows in each stage of the reduction phase is calculated by the formula

$$r_{i+1} = 2[r_i/3] + r_i \bmod 3$$

If $r_i \bmod 3 = 0$, then $r_{i+1} = 2r_i/3$

If the value calculated from the above equation for number of rows in each stage in the second phase and the number of rows that are formed in each stage of the second phase does not match, only then the half adder will be used. The final product of the second stage will be in the height of two bits and passed on to the third stage. During the third stage

the output of the second stage is given to the carry propagation adder to generate the final output.

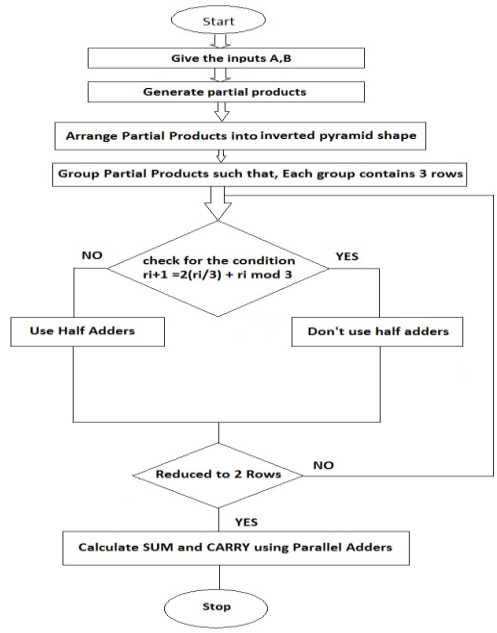
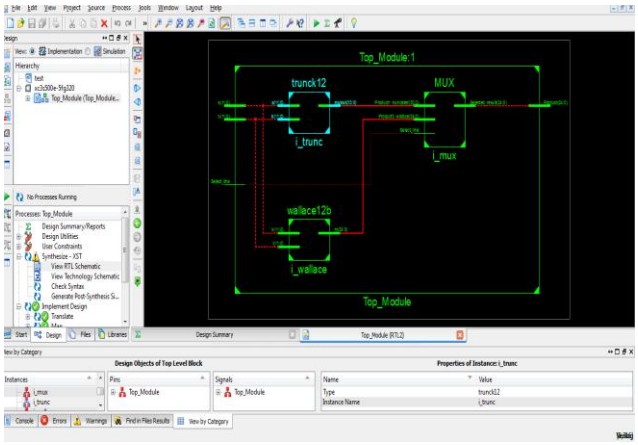


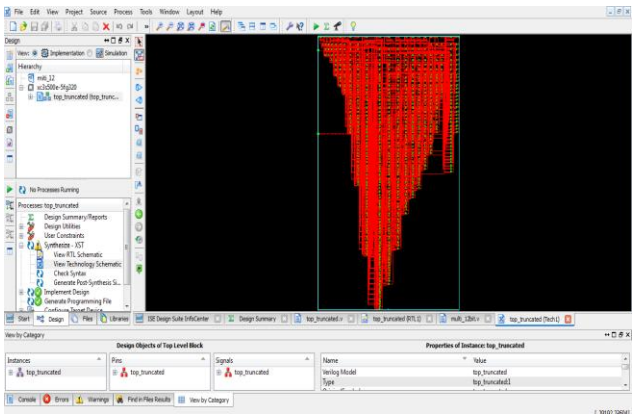
Fig 4: Modified Dada Flow Chart

III. RESULTS

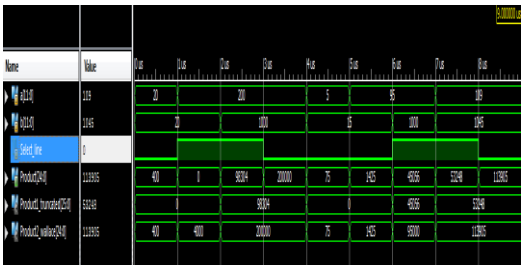
RTL schematic



Technological schematic



Simulated Waveform



Comparison Table

Multiplier	No .of LUT's	Delay	Memory
Truncated with modified Dada	618	30.679ns	245 MB
Truncated with array	648	41.102ns	271 MB

IV. CONCLUSION

Here, in this project two different multipliers are designed which are array multiplier and modified Wallace multiplier along with the combination of truncated multiplier. In the proposed design which is nothing but truncated with modified Wallace the area (in terms of LUT's) is less which are 618 when compare to the existing truncated with array multiplier which are 648. So obviously the power is also reduced because it is calculated based on the number of LUT's. At the same time the delay and memory requirements for the proposed design is better when compare with the existed design .This multipliers output are derived depending on multiplexer selection line, which depends on the user. In future based on the requirements there may be a chance to change the multipliers.

V. REFERENCES

[1] Shen-Fu Hsiao, Jun-Hong Zhang Jian, and Ming-Chih Chen," Low-Cost FIR Filter Designs Based on Faithfully Rounded Truncated Multiple Constant Multiplication/Accumulation" ieee transactions on circuits and systems—ii: express briefs, vol. 60, no. 5, may 2013.

[2] M. M. Peiro, E. I. Boemo, and L. Wanhammar, "Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.,vol. 49, no. 3, pp. 196–203, Mar. 2002.

[3] C.-H. Chang, J. Chen, and A. P. Vinod, "Information theoretic approach to complexity reduction of FIR filter design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 8, pp. 2310–2321, Sep. 2008.

[4] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution—A new approach to versatile subexpressions sharing in multiple constant multiplications,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 2,pp. 559–571, Mar. 2008.

[5] F. Xu, C. H. Chang, and C. C. Jong, "Contention resolution algorithms for common subexpression elimination in digital filter design," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 52, no. 10, pp. 695–700, Oct. 2005.

[6] I.-C. Park and H.-J. Kang, "Digital filter synthesis based on an algorithm to generate all minimal signed digit representations," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 21, no. 12, pp. 1525–1529, Dec. 2002.

[7] C.-Y. Yao, H.-H. Chen, T.-F. Lin, C.-J. J. Chien, and X.-T. Hsu, "A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 11,pp. 2215–2221, Sep. 2004.

[8] O. Gustafsson, "Lower bounds for constant multiplication problems," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 11, pp. 974–978, Nov. 2007.

[9] Y. Voronenko and M. Puschel, "Multiplierless multiple constant multiplication," ACM Trans. Algorithms, vol. 3, no. 2, pp. 1–38, May 2007.

[10] D. Shi and Y. J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders," IEEE Trans. Circuits Syst.I, Reg. Papers, vol. 58, no. 1, pp. 126–136, Jan. 2011.

[11] P. K. Meher, "New approach to look-up-table design and memory-based realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 57, no. 3, pp. 592–603, Mar. 2010.

[12] P. K. Meher, S. Candrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,"IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul. 2008.

[13] S. Hwang, G. Han, S. Kang, and J.-S. Kim, "New distributed arithmetic algorithm for low-power FIR filter implementation," IEEE Signal Process.Lett., vol. 11, no. 5, pp. 463–466, May 2004.

[14] H.-J. Ko and S.-F. Hsiao, "Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation,and rounding," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 5, pp. 304–308, May 2011.

